AD-A066 444 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/2

A LAND MANAGEMENT MODEL USING DANTZIG-WOLFE DECOMPOSITION.(U)

DEC 78 L NAZARETH N00014-75-C-0267

UNCLASSIFIED SOL-78-31 NL

1 OF 1

AD A066444

END

DATE FILMED

5-79

DDC

LEVEL II

(12)

Systems
Optimization
Laboratory

Department of Operations Research
Stanford University
Stanford, CA 94305

LEVEL II

(12)

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305

(6) A LAND MANAGEMENT MODEL USING DANTZIG-WOLFE DECOMPOSITION.

by

(10) L. Nazareth

(9) TECHNICAL REPORT SOL 78-31
December 1978

(11) Dec 78 (14) SOL-78-31

(12) 38p.

(15)

Research and reproduction of this report were partially supported by the Office of Naval Research Contract N00014-75-C-0267; the National Science Foundation Grants MCS76-20019 A01 and ENG77-06761 A01; and the Department of Energy Contract EY-76-S-03-0326 PA #18.

408765

LB

Abstract

This paper deals with a mathematical model designed to provide guidelines for managing a land resource over an extended period of time.

We develop a framework which permits sequences of management decisions to be conveniently formulated, and their associated costs and benefits specified. This takes the form of a network. Each path in the network represents a possible decision sequence. We study how to select suitable decision sequences and what proportion of the resource to manage with each selected sequence, so as to optimize some specified objective and meet the constraints imposed on management of the resource. An L.P. model is formulated. The solution strategy decomposes the L.P. matrix using Dantzig-Wolfe decomposition and solves the subproblems efficiently by dynamic programming or a network flow algorithm. Computational aspects are discussed and the concepts and procedures are illustrated in the Appendix, for forest management.

This paper is a substantially revised version of Reference [6].

Acknowledgments

I am much indebted to Daniel I. Navon who introduced me to the subject and provided the initial motivation and lots of help along the way. My warmest thanks also go to Dr. Stuart Dreyfus for his encouragement, guidance and advice.

I am grateful for the many useful discussions I have had with Harry Jansen and Andres Weintraub. Also, thanks to Shail Parikh for discussions on his reservoir model.

Finally, many thanks to Lynda Schrotenboer for typing the manuscript.

A LAND MANAGEMENT MODEL USING DANTZIG-WOLFE DECOMPOSITION*

by

L. Nazareth

1. Introduction

In recent years many land management models have been formulated, e.g., Navon, et al. [1,2], RCS [3], Heady and Chandler [4], Shoemaker [5]. One of the most broadly-applicable formulations appears to be the dynamic linear programming model. Here the equations and inequalities are linear, and the term dynamic reflects an essential feature of the type of management under consideration -- that management decisions have to be made during each of a sequence of successive time intervals which span the period of planning. Our model is of this type.

The total resource to be managed is assumed to be specified as a set of resource classes, denoted by $\mathbb{C}^1, \mathbb{C}^2, \ldots, \mathbb{C}^k$, each capable of yielding one or more quantifiable products, either simultaneously or sequentially; a resource class is obtained by grouping together those portions of the total resource which have similar initial conditions, productive potential, economic characteristics and response to management. At the level at which planning is being conducted it is assumed to be reasonable to consider such an aggregate of land

*This paper is a substantial revision of Reference [6].

parcels as a homogenous entity for which a given set of management alternatives would be explored cf. for example, the timber model of Navon, et al. [1].

The period of planning, over which management plans are to be developed, is subdivided into a set of time intervals. These planning intervals are usually of equal length, and span the planning period.

Consider a particular resource class $\mathbb{C}^k$. During each planning interval a management or control decision may be carried out, e.g., on a timber class a control decision may specify what proportion of standing timber should be removed during the interval. A control decision results in (i) cost expenditure, (ii) the realization of benefits, e.g., agricultural produce, timber, forage, etc., (iii) the development or degradation of the resource class. A sequence of control decisions carried out on a portion of the resource class over the span of the planning period therefore implies a flow of costs and benefits. We shall call this decision sequence and its associated costs and benefits a <u>management alternative</u>. In general, for each resource class, there will be many such alternatives each having a different impact on the resource. Note that specifying a management alternative is independent of the total acreage to which it is applied. Thus one of the assumptions inherent in the model is the linearity of costs and benefits of each management alternative as a function of acreage managed.

We shall denote the j-th management alternative for class $\mathbb{C}^k$ by the pair of vectors $(\underset{\sim}{M}_j^k, \underset{\sim}{B}_j^k)$, where $\underset{\sim}{M}_j^k$ is a vector whose i-th component M_{ij}^k is the cost per unit area of management decisions in interval i; similarly B_{ij}^k is the benefit per unit area in interval i. We shall denote the area, i.e., number of acres of $\mathbb{C}^k$ managed by $(\underset{\sim}{M}_j^k, \underset{\sim}{B}_j^k)$ by x_j^k. Our problem can then be stated as follows: "Determine x_j^k $\forall k,j$ sc that all constraints are satisfied and some specified objective is optimized."

Because the number of management alternatives is potentially very large, we develop a convenient way to specify them, and their associated costs and benefits. This takes the form of a network. Next we discuss possible constraints and objectives and develop the L.P. model. Finally, we describe a solution strategy which employs Dantzig-Wolfe decomposition coupled with dynamic programming or a network flow algorithm for efficient solution of subproblems.

2. <u>Specifying the Management Alternatives</u> (see also Appendix I)

It is assumed that at the beginning of each interval, any part of resource class $\mathbb{C}^k$ can exist in one of only a finite set of distinguishable states. We shall represent the states of class $\mathbb{C}^k$ at the beginning of the i-th interval by C^k_{ij} where $j = 1, 2, \ldots, n_i$. C^k_{ij} will often be specified by abstracting certain significant characteristics of class $\mathbb{C}^k$ called state parameters, and a state C^k_{ij} is determined by specifying the value that each parameter can take. For the purposes of management and at the level at which management is being conducted, the characteristics chosen must adequately describe the productivity of class $\mathbb{C}^k$.

Suppose some part of class $\mathbb{C}^k$ is in state C^k_{ip} at the beginning of interval i. The process of management during interval i is now seen to consist of converting this portion, in state C^k_{ip}, into one of the states $C^k_{(i+1)j}$, $j = 1, 2, \ldots, n_{(i+1)}$, which are allowable at the end of interval i. This state transformation is achieved by designing and implementing a suitable control decision. State transformations between certain pairs of states may not be possible, or may be excluded as being undesirable. The process of defining states and state transformations is clearly an iterative one, since they interact with one another.

Each state transformation $C^k_{ip} \rightarrow C^k_{(i+1)q}$, achieved by a control decision carried out during the i-th interval, incurs certain costs and yields one or more benefits. Costs and benefits are functions of

the pair of states involved in the transformation and the control decision by which it is achieved. These are specified on a per unit area basis.

Finally, if the states and transformations for each interval are displayed simultaneously, we obtain a directed network in which each path from the initial state to any of the states at the end of the planning period defines a management alternative. Within this framework a variety of planning decisions may be made simply by pruning the network.

In practice, a number of simplifying situations arise. (a) In many cases it will be possible to use the same set of states and transformations for every interval, subject only to discounting dollar value of cost and benefit. (b) The states of one interval may be deducible from the states of a previous interval. (c) The calculation of costs and benefits may be possible from the state parameters and some basic knowledge about the control operations involved. (d) Different resource classes may share the same network, but with each having a different initial state.

In this way, the set of networks that are needed, may be generated from a much smaller information base.

In summary, the main steps are:

(1) Determine the resource classes.

(2) Determine the planning intervals.

(3) Determine the states for each resource class.

(4) Determine the state transformations for each resource class.

(5) Determine costs and benefits for each transformation.

(6) Develop the network for each class.

In Nazareth [6], Chapter 4, application of the above procedure to specific resources is studied, and points (a)-(d) above are illustrated. Specifically, different methods for discretizing the state space for a timber resource are developed, and the way to include reforestation is considered. The above approach has also been successfully applied to range land management (see Jansen [7]).

3. Examples of Constraints and Objectives

3.1. Notation

In addition to the notation introduced in Section 1, we use the following:

P -- The number of planning intervals.

R -- The number of resource classes.

S^k -- The set of all alternatives for $\mathbb{C}^k$. Assume that there are N^k of these. For the purpose of stating the constraints we shall act as though all alternatives are generated explicitly from the network. However, the solution strategy works directly on the network, as will be discussed in Section 4.

3.2. Resource Class Constraints

3.2.1. Variables definition rows

$$\sum_{j=1}^{N^k} M_{ij}^k \, x_j^k = X_i^k \; ; \qquad \sum_{j=1}^{N^k} B_{ij}^k \, x_j^k = Y_i^k \tag{3.1}$$

$i = 1, \ldots, P$; $k = 1, \ldots, R$, X_i^k represent the total cost incurred in interval i for class k. Y_i^k represents the total benefit accrued in interval i for class k. These are known as the *return variables* of class $\mathbb{C}^k$. We shall call x_j^k the principal variables.

3.2.2. Availability constraints

(a) The total amount managed by all alternatives of class k is constrained to be less than or equal to the total area available, a^k.

$$\sum_{j=1}^{N^k} x_j^k \leq a^k \qquad k = 1, \ldots, R \, . \tag{3.2}$$

(b) For the first m^k intervals, class k may not be totally accessible. Say only $P_i^k\%$ of the total quantity a^k is accessible for all intervals such that $1 \leq i \leq m^k$.

$$\sum_{i \in J_i^k} x_j^k \leq \frac{P_i^k}{100} a^k \qquad \begin{array}{l} 1 \leq i \leq m^k \\ k = 1, \ldots, R \end{array} \tag{3.3}$$

where J_i^k is the set of all alternatives from S^k which involve some nontrivial management of class $\mathbb{C}^k$ in intervals prior to the i-th.

(c) A variety of additional constraints may be imposed on class k, limiting the amount of land that can be in a particular state or the amount that can be transformed from our state into another, or requiring that at least a certain amount of land be in a particular state during a given interval. Such constraints are important when it becomes necessary to ensure that side benefits, not directly specified, e.g., recreation value, are maintained at an acceptable level.

3.3. Global Constraints

3.3.1. Periodic constraints

These constrain the total cost incurred in interval i and the total benefit accrued in interval i and may be specified for each class or globally over all classes.

$$X_i^k \leq U_i^k \quad \text{or} \quad \sum_k X_i^k \leq U_i$$

(3.4)

$$Y_i^k \geq V_i^k \quad \text{or} \quad \sum_k Y_i^k \geq V_i \ .$$

The components of U_i^k represent the upper bounds on each type of cost incurred in managing $\mathbb{C}^k$ and the components of V_i^k the minimum amount of each type of benefit to be produced. Similarly for U_i and V_i except the constraint is now over all classes.

3.3.2. Flow constraints

These for example constrain the total benefit derived in interval i to be within certain levels of the benefit accrued in the previous interval, e.g.,

$$(3.5) \quad \sum_k Y_i^k \leq (1 + u) \sum_k Y_{i-1}^k \quad \text{and} \quad \sum_k Y_i^k \geq (1 - \ell) \sum_k Y_{i-1}^k \ .$$

See [1] for examples of such constraints for timber.

3.4. Example of Objective

Objectives will be of the form, for example, minimize total cost

$$\text{(3.6)} \qquad \text{Minimize} \sum_{i=1}^{P} \sum_{k=1}^{P} X_i^k/(1 + r)^i$$

where r is the interest rate.

One form of the L.P. matrix derived from the previous equations is illustrated, for three resource classes in Figure 3.1. Each submatrix D^k corresponds to a network whose paths determine the column of coefficients $(M_{ij}^k, B_{ij}^k)^T$ of the principal variables associated with class $\mathfrak{C}^k$.

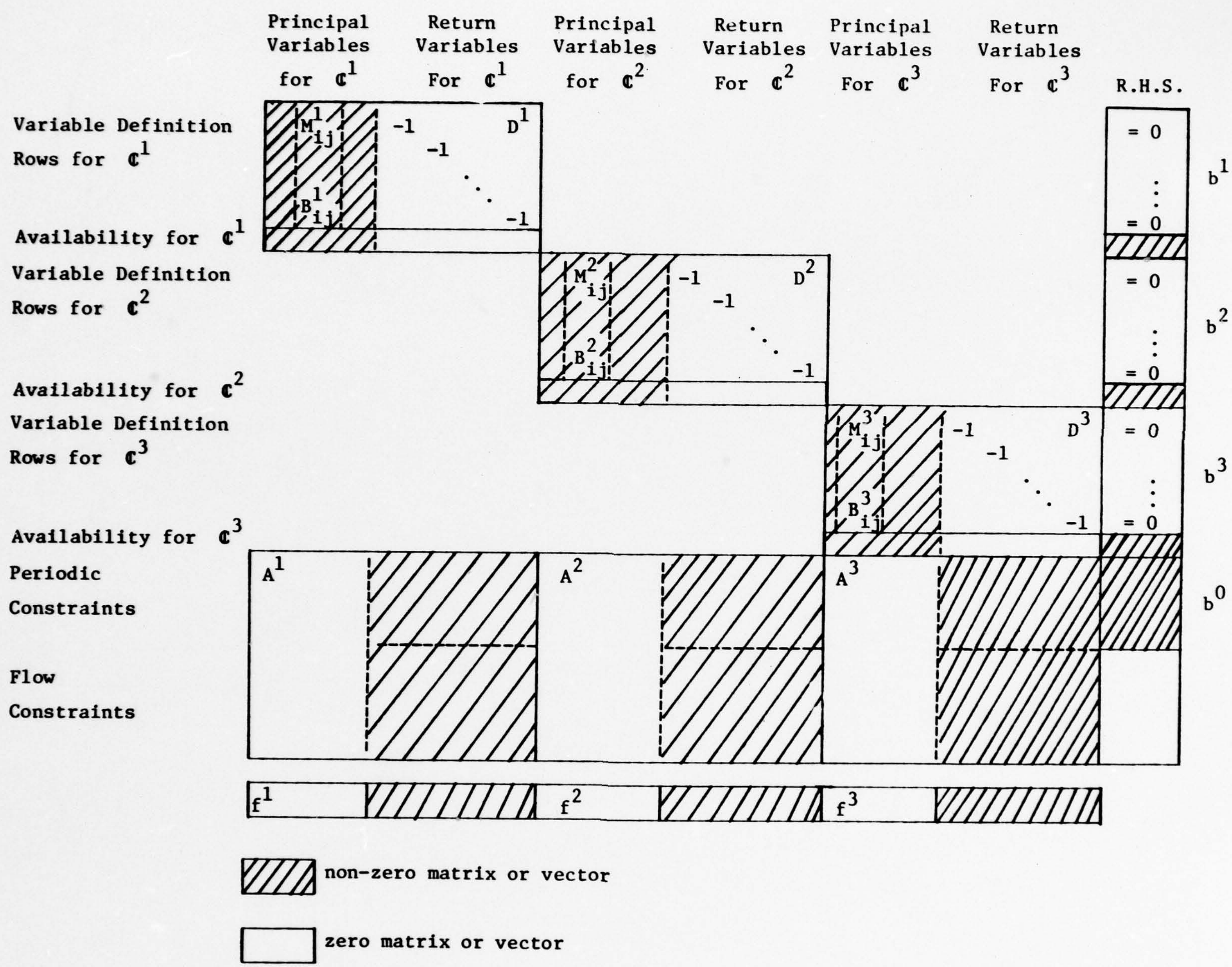
Principal Variables for $\mathbb{C}^1$
Return Variables For $\mathbb{C}^1$
Principal Variables for $\mathbb{C}^2$
Return Variables For $\mathbb{C}^2$
Principal Variables For $\mathbb{C}^3$
Return Variables For $\mathbb{C}^3$
R.H.S.
Variable Definition Rows for $\mathbb{C}^1$
Availability for $\mathbb{C}^1$
Variable Definition Rows for $\mathbb{C}^2$
Availability for $\mathbb{C}^2$
Variable Definition Rows for $\mathbb{C}^3$
Availability for $\mathbb{C}^3$
Periodic Constraints
Flow Constraints
M^1_{ij}
B^1_{ij}
M^2_{ij}
B^2_{ij}
M^3_{ij}
B^3_{ij}
D^1
D^2
D^3
-1
= 0
b^1
b^2
b^3
b^0
A^1
A^2
A^3
f^1
f^2
f^3
non-zero matrix or vector
zero matrix or vector

Figure 3.1

4. Choice of the Solution Strategy

The various classes $\mathbb{C}^k$, $k = 1, \ldots, R$ compete with one another for the resources available. From the set S^k, $\forall k$ we must select that combination of alternatives and their levels which "best" meet the constraints of the analytic model in Section 3. This optimal solution would identify which classes to manage intensively, which classes it would be profitable to convert from one mode of production to another, and when to carry out such a conversion.

The most obvious technique of solution is to explicitly generate each of the submatrices D^k and thus to develop the complete L.P. matrix. This will only be feasible for networks with a relatively small number of paths.

4.1. The Decomposition Model

We now seek to develop a technique which avoids an explicit generation of each alternative. In this technique we decompose the matrix into a master and a set of subproblems using Dantzig-Wolfe Decomposition. Each subproblem has a network associated with it, and, as we shall see, the nature of the subproblem constraints and objective are such that it may be efficiently solved, for example by Dynamic Programming on this network. The rows of Figure 3.1 corresponding to the Periodic and Flow Constraints and the Objective define the master, and the rows corresponding to the Variable Definition and Availability rows for class $\mathbb{C}^k$, define the k-th subproblem.

Using the notation developed in Figure 3.1, we may write the analytic model as:

$$
\begin{array}{llr}
 & \text{Minimize } \underset{\sim}{f}^1 \underset{\sim}{z}^1 + \underset{\sim}{f}^2 \underset{\sim}{z}^2 + \cdots + \underset{\sim}{f}^R \underset{\sim}{z}^R & \\
 & \text{Subject to } A^1 \underset{\sim}{z}^1 + A^2 \underset{\sim}{z}^2 + \cdots + A^R \underset{\sim}{z}^R = b^0 & \\
 & D^1 \underset{\sim}{z}^1 \qquad\qquad\qquad\qquad = \underset{\sim}{b}^1 & \\
(4.1) & \qquad\quad D^2 \underset{\sim}{z}^2 \qquad\qquad = \underset{\sim}{b}^2 & \\
 & \qquad\qquad\qquad\qquad D^R \underset{\sim}{z}^R = \underset{\sim}{b}^R & \\
 & \underset{\sim}{z}^1, \underset{\sim}{z}^2, \ldots, \underset{\sim}{z}^R \geq 0 \, . &
\end{array}
$$

where $\underset{\sim}{z}^k$ is the vector of principal and return variables $\underset{\sim}{z}^k = (\underset{\sim}{x}^k, \underset{\sim}{X}^k, \underset{\sim}{Y}^k)$ the vectors $\underset{\sim}{f}^1, \ldots, \underset{\sim}{f}^R$ have zeros in the positions corresponding to the principal variables and the columns of $A^1, \ldots, A^R$ corresponding to the principal variables are also zero (cf Figure 3.1). Denote by $\hat{f}^i$ and $\hat{A}^i$ respectively the elements and columns of f^i and A^i corresponding only to the return variables.

Using the standard Dantzig-Wolfe Decomposition principle [10], [12], the k-th subproblem is then of the form

$$\text{Minimize} \quad (\hat{f}^k - \underset{\sim}{\Pi}\hat{A}^k)\begin{pmatrix} \underset{\sim}{X}^k \\ \underset{\sim}{Y}^k \end{pmatrix} - \Pi_{ok}$$

$$\text{(4.2)} \qquad \text{Subject to} \quad D^k \underset{\sim}{Z}^k = \underset{\sim}{b}^k$$

$$\underset{\sim}{Z}^k \geq 0$$

where $\underset{\sim}{\Pi}$ are the dual variables (prices) assorted with the non-convexity constraints of the master problem, and Π_{ok} corresponds to the convexity constraint arising from the k-th subproblem.

4.2. Solution of Subproblems

4.2.1. Basic considerations: Dynamic programming solution

In order to explain the solution strategy in a simple context, let us consider in Figure 3.1 only availability constraints of the form (3.1). Assume, also for simplicity, a single cost and a single benefit per transformation, and let $\underset{\sim}{\mu}^k = (\underset{\sim}{\hat{f}}^k - \Pi\underset{\sim}{\hat{A}}^k)$. Henceforth we drop superscript k, to simplify the notation.

The k-th subproblem above may be written as:

$$\text{Minimize} \quad \underset{\sim}{\mu} \cdot \begin{pmatrix} \underset{\sim}{X} \\ \underset{\sim}{Y} \end{pmatrix} - \Pi_{ok}$$

$$\text{Subject to} \quad \sum_j M_{ij} x_j - X_i = 0 \qquad \forall\, i$$

$$(4.3) \qquad \sum_j B_{ij} x_j - Y_i = 0 \qquad \forall\, i$$

$$\sum_j x_j = a$$

$$x_j \geq 0 \ .$$

Partitioning $\underset{\sim}{\mu}$ into $(\tilde{\underset{\sim}{\mu}}, \hat{\underset{\sim}{\mu}})$ where: $\tilde{\underset{\sim}{\mu}}$ corresponds to the vector $(\underset{\sim}{X} = X_1, \ldots, X_i, \ldots)$ and $\tilde{\underset{\sim}{\mu}}$ corresponds to the vector $(\underset{\sim}{Y} = Y_1, \ldots, Y_i, \ldots)$ the subproblem objective may then be written as follows:

$$\text{Minimize} \quad \sum_j \left[\sum_i (\tilde{\mu}_i M_{ij} + \hat{\mu}_i B_{ij}) \right] x_j - \Pi_{ok}$$

$$(4.4) \qquad \text{Subject to} \quad \sum_j x_j = a$$

$$x_j \geq 0 \ .$$

The minimum value can always be attained for such a problem by a single alternative, say x_t at level a and all other alternatives x_j, $j \neq t$ at level zero; t will be such that $\sum_i (\tilde{\mu}_i M_{it} + \tilde{\mu}_i B_{it})$ is a minimum. Now recall that (M_{it}) and (B_{it}) are obtained from the i-th path in the network corresponding to class $\mathbb{C}$. Therefore solving the subproblem corresponds to finding the path of minimum length in the network for $\mathbb{C}$ whose arc values have beem computed as follows:

-- The cost in interval i is multiplied by $\tilde{\mu}_i$ and added to the benefit multiplied by $\tilde{\mu}_i$.

This minimum path can be found using the recursive technique of Dynamic Programming [8].

Once an alternative giving the minimum value of the subproblem objective for each subproblem has been found for the current set of master prices, the coordinates of the extreme point corresponding to the return variables must be returned to the master. These are given by the vector $\underline{v} = (\underset{\sim}{M}_t\ a,\ \underset{\sim}{B}_t\ a)$ and the appropriate column of the master for non-convexity rows is now generated from $\hat{A}\ \underline{v}$.

4.2.2. <u>Extensions to solution strategy</u>

We shall illustrate this for two cases:

(1) <u>Extending the DP when adding constraints of the form (3.3) of Section 3.2.2</u>. It will suffice to consider a single constraint of the form

$$\sum_{i \in J_1^k} x_1^k = \frac{P_1^k}{100} a^k \tag{4.5}$$

where again, we shall drop the superscript k for simplicity. (Thus $(1 - P_1)\%$ of the resource class is not accessible during the first planning interval.) Consider $\mathbb{C}$ to be partitioned into two sets whose acreages are in the ratio $P_1:(1 - P_1)$. Recall that the initial state is represented by C_{11}, and assume that acreage in state C_{11} transforms into state C_{2t} at the end of the interval 1 under no management (which may mean just some basic maintenance). Then the solution to the subproblem corresponding to resource class $\mathbb{C}$ is obtained by finding the minimum paths in the network modified as described earlier in Section 4.2.1.

(i) From C_{11} to one of the final states.

(ii) From C_{11} through C_{2t} to one of the final states.

These can be found simultaneously in one pass of the DP algorithm. If these paths or management alternatives are given by

$$\begin{pmatrix} \underline{M}_v \\ \underline{B}_v \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \underline{M}_w \\ \underline{B}_w \end{pmatrix}$$

respectively, then the components of the extreme point corresponding to return variables are

$$P_1 \underset{\sim}{M}_v + (1 - P_1)M_w$$

and these are given back to the master for the next iteration of the decomposition algorithm. The reader will have no difficulty extending this scheme to several accessibility constraints of the form (4.5).

(2) Using a network flow algorithm to handle constraints discussed in 3.2.2(c). This is best described by reformulating the resource class constraints of Section 3.2 as follows:

-- Let $c_{i,pq}$, $b_{i,pq}$ and $z_{i,pq}$ represent the cost, benefit, and acreage, respectively of the transformation $C_{ip} \to C_{(i+1)q}$ (where again superscripts k are dropped for convenience). Then we can write (3.1) as

(4.6) $$\sum_{p,q} c_{i,pq}\, z_{i,pq} = X_i\ , \qquad \sum_{p,q} b_{i,pq}\, z_{i,pq} = Y_i$$

Availability constraints of Section 3.2.2 can be specified by associating upper and lower bounds with each arc of the network and by introducing dummy nodes and arcs in the usual manner. Then a subproblem can be solved using a network flow algorithm, whose objective function is given, as in (4.3) by $\underset{\sim}{\mu} \cdot \binom{\underset{\sim}{X}}{\underset{\sim}{Y}}$, with $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ defined by (4.6); the components of the extreme point corresponding to the return variables are deduced from the optimal flows in the network and returned to the master problem, for the next iteration of the decomposition algorithm.

4.2.3. Other related model structures

Dantzig-Wolfe decomposition coupled with dynamic programming (DP) is, of course, not new. Other models which employ this solution strategy are, for example, those of Dzielinski and Gomory [16] and Parikh [14]; these models share some features in common with ours. However, the particular manner in which this strategy is formulated and used differs substantially from one model to another.

The model of Dzielinski and Gomory [16] is a particular example of the production scheduling problem formulated in Lasdon [11, p. 171]. The L.P. constraint matrix has the form given in Figure 4.1, where θ_{ij} corresponds to the j-th schedule or activity (selected from a set, say S_i, defined by the 'technological constraints') for producing item i, where $i = 1, 2, \ldots, I$. The problem is to select a particular subset of schedules so as to satisfy the resource constraints and minimize the cost of operating all activities over time.

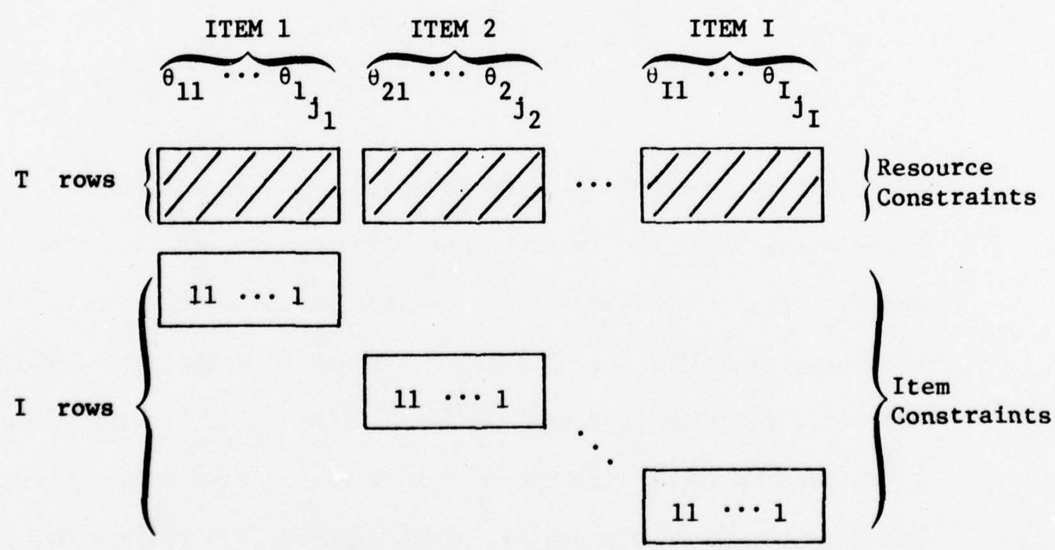

Figure 4.1

When $I >> T$ most blocks will have only one $\theta_{ij} \neq 0$ (and therefore $= 1$) and the L.P. approximates the underlying integer programming problem. The item constraints defined a single decomposed subproblem (giving one convexity row in the master). This is separable into I Wagner-Whitin type deterministic inventory problems which can be solved by DP. The constraint matrix is similar to the special case discussed in Section 4.2.1, when return variables are substituted out. The manner in which dynamic programming is used to select schedules is quite different from our use of it, and there is no analogue to the extensions discussed in Section 4.2.2. Note also that the model of Dzielinksi and Gomory is concerned with detailed plans for production scheduling.

Our model also shares features in common with the model of Parikh [14] for long range operation of a multiple water reservoir system. Each reservoir system plays a role analogous to our resource class, and return variables analogous to ours are defined for water release and on-peak and off-peak energy production, for each reservoir and time period. The constraint matrix is also block angular (cf Figure 3.1), with the return variables coupling the constraints for each reservoir to the constraints for the integrated system. Each set of constraints for a reservoir system defines a subproblem, which is solved by a particular application of DP over a grid defined by discretizing storage and water release. In contrast to our model, note that there is no partitioning of a resource class, each partition to be managed by a different alternative, and that the considerations of Section 4.2.2 again do not have a direct analogue.

Amidon and Akin [9] have used dynamic programming for forest management. Tcheng [13] has developed a decomposition model for forest management. Our model combines and generalizes these approaches. We have drawn rather directly upon the concepts of optimal control theory, in order to set up a framework for specifying management alternatives for each resource class. This would normally be done with the cooperation of a resource manager who is familiar with its characteristics. Thus we have structured the process of defining what can be done to a resource class. We have then shown how these resource classes can be combined in a block angular L.P. model, to which the Dantzig-Wolfe decomposition principle can be applied. In the next section, we describe the design of a computer system which implements these ideas in a general way.

5. Computational Aspects, Discussion of Experimental Computer Program, Computational Results

Figure 5.1 below is a broad-brush description of the information requirements and possible options, in a system designed to implement the resource allocation model and the solution strategy developed in this paper. The information needed to specify a network of management alternatives for any resource class is as follows:

-- The states of each interval.

-- State transformations for each interval.

-- Costs associated with each transformation.

-- Benefits associated with each transformation.

Each resource class, however, has its own special characteristics. These may make possible the calculation of the above information from a much smaller base of information. For some classes the direct approach of reading in matrices that specify states, transformations, costs and benefits may be best; for other classes it may be preferable to generate these items of information from data that is more natural and more compact. See Nazareth [6] Chapter 4, for examples of this. A system for specifying networks which is built around general I/O routines, should therefore be flexibly designed so as to incorporate special features of a resource class.

In the experimental computer program we implemented the portion of Figure 5.1 given by the shaded boxes and the joining paths marked with double arrows.

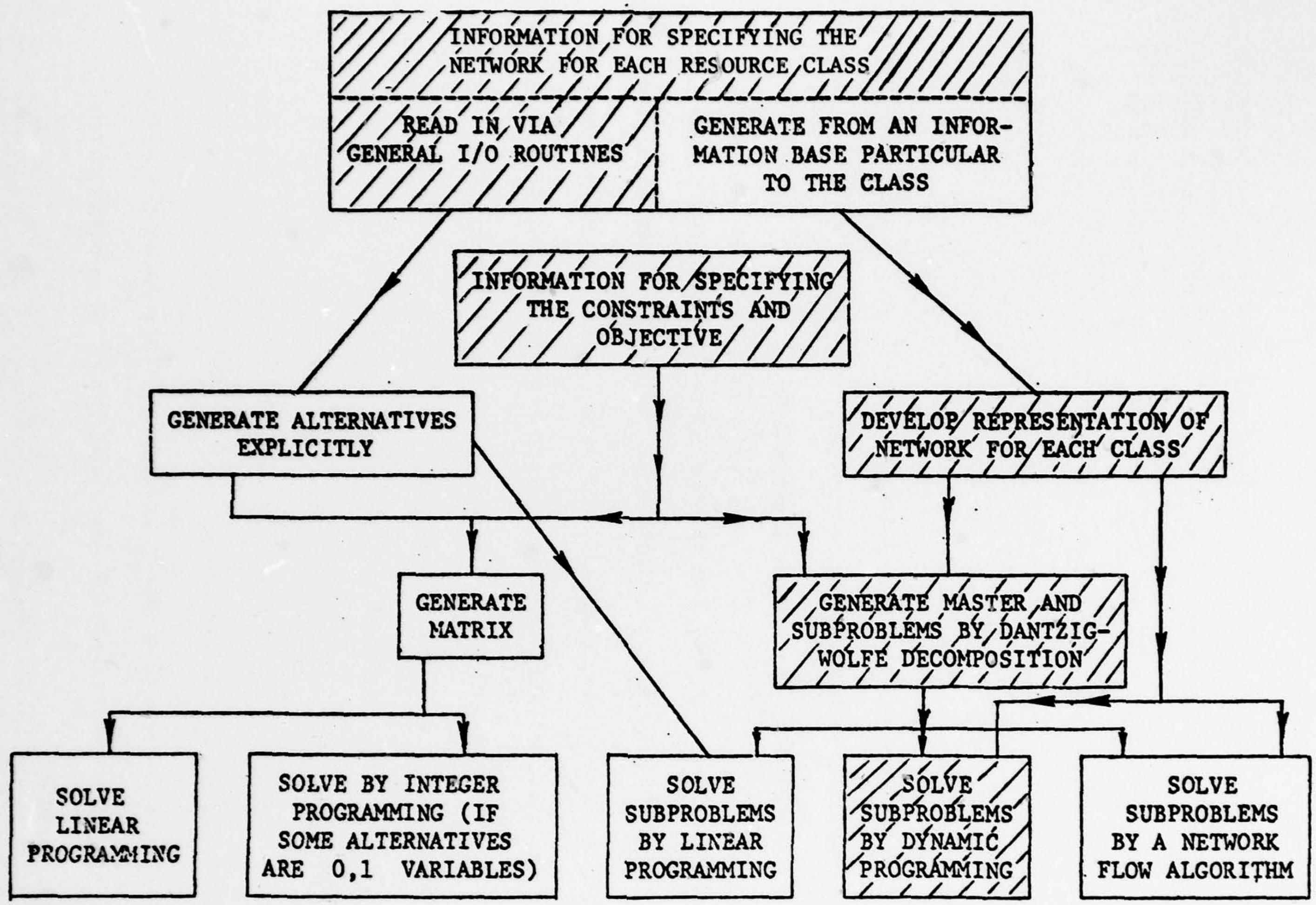
INFORMATION FOR SPECIFYING THE NETWORK FOR EACH RESOURCE CLASS
READ IN VIA GENERAL I/O ROUTINES
GENERATE FROM AN INFORMATION BASE PARTICULAR TO THE CLASS
INFORMATION FOR SPECIFYING THE CONSTRAINTS AND OBJECTIVE
GENERATE ALTERNATIVES EXPLICITLY
DEVELOP REPRESENTATION OF NETWORK FOR EACH CLASS
GENERATE MATRIX
GENERATE MASTER AND SUBPROBLEMS BY DANTZIG-WOLFE DECOMPOSITION
SOLVE LINEAR PROGRAMMING
SOLVE BY INTEGER PROGRAMMING (IF SOME ALTERNATIVES ARE 0,1 VARIABLES)
SOLVE SUBPROBLEMS BY LINEAR PROGRAMMING
SOLVE SUBPROBLEMS BY DYNAMIC PROGRAMMING
SOLVE SUBPROBLEMS BY A NETWORK FLOW ALGORITHM

Figure 5.1

(i) The information needed to specify each network is fed in as follows:

For each state parameter a list is specified of all the levels it can assume. A state is defined by specifying a level for each state parameter. Thus each state may be identified by a set of indices. These determine the position within each list of the corresponding parameter levels.

The state transformations, costs and benefits are specified as 2-dimensional data arrays or tables with as many rows or columns as there are states. The (i,j) element of, for example, the cost table gives the cost of the transformation that converts the i-th state to the j-th state. When states vary from interval to interval, each of the above three tables contain many imbedded zeros. However, we have adopted this approach of having the same three tables apply to every interval rather than that of having three tables, each of a smaller dimension, per interval, for ease of implementation.

A complete example which illustrates the above material and that of the preceding sections is given [6].

(ii) The information needed for the master constraints (see Figure 3.1) and other problem parameters are flexibly specified via the NAMELIST option of Fortran.

(iii) For the subproblems only availability constraints of the form (3.2) in Section 4, were implemented in the experimental program. In implementing the Decomposition -- D.P. solution

strategy, particular care was taken in the design of the dynamic programming subroutine; in particular, explicitly indexing within 2 or higher dimensional data arrays avoids a large number of multiplications involved in locating array elements.

(iv) Program validation was carried out by comparing the results with those obtained by explicitly listing alternatives for a simple example and solving by the Simplex Method.

The preliminary results concerning the efficiency of the solution strategy have been encouraging. The solution strategy has been tested on a case study involving the management of range resource classes reported in [6]; further studies for timber and combinations of timber and range management are planned, and will be reported in more detail at a later date.

6. Conclusion

Our aim in this paper has been to develop a flexible planning tool for laying down broad guidelines on resource management. Its purpose is to help resolve questions which arise in multiple use management of a land resource, e.g., those discussed at the beginning of Section 4. Although we can only speculate at this point, we feel that in addition to potential uses already discussed for forest, range, wildland and agricultural management, these ideas may carry over to land reclamation and strip mine management [17] and to pest-control management [15].

References

[1] Navon, D.I., "Timber RAM ... A Long Range Planning Method for Commercial Timberlands Under Multiple-Use Management," USDA Forest Service Research Paper, PSW Forest and Range Experiment Station, Berkeley (1970-71).

[2] Navon, D.I., et al., TIMBER RAM USERS MANUAL, Parts I, II, III and IV, U.S. Department of Agriculture, PSW Forest and Range Experiment Station, Berkeley (1972).

[3] RESOURCE CAPABILITY SYSTEM ... A USER'S GUIDE, Forest Service -- U.S. Department of Agriculture, PSW Forest and Range Experiment Station, Berkeley (1972).

[4] Heady, C.O., and W. Chandler, LINEAR PROGRAMMING METHODS, Iowa State College Press, Ames, Iowa, 597 pages (1958).

[5] Shoemaker, C., "Optimization of Agricultural Pest Management," Department Environmental Engineering and Entomology, Cornell University (1972).

[6] Nazareth, J.L., "A Resource Allocation Model for Land Management and A Solution Strategy that Combines Dantzig-Wolfe Decomposition and Dynamic Programming," Rep. ORC 73-20 (1973).

[7] Jansen, H.E., "A Range Resource Allocation Method," (Ph.D. Thesis), School of Forestry, University of California, Berkeley (1973).

[8] Bellman, R.E. and S.E. Dreyfus, APPLIED DYNAMIC PROGRAMMING, Princeton University Press, Princeton, N.J.

[9] Amidon, E., and G. Akin, "Dynamic Programming to Determine Optimum Levels of Growing Stock," Forest Science, Vol. 14, No. 3 (1968).

[10] Dantzig, G.B., LINEAR PROGRAMMING AND EXTENSIONS, Princeton University Press, Princeton, N.J., 632 pages.

[11] Lasdon, L., OPTIMIZATION THEORY FOR LARGE SYSTEMS, The MacMillan Co. (1970).

[12] Dantzig, G.B. and P. Wolfe, "The Decomposition Algorithms for Linear Programs," Econometrica, Vol. 29, No. 767 (1961)

[13] Tcheng, T.H., "Scheduling of Large Forestry Cutting Problems by Linear Programming Decomposition," (Ph.D. Thesis), University of Iowa (1966).

[14] Parikh, S.C., "Linear Dynamic Decomposition Programming of Optimal Long-Range Operation of a Multiple Multi-purpose Reservoir System," ORC 66-28, Operations Research Center, University of California, Berkeley (1966); (also published in Proceedings of 4th International Conference on Operations Research, Boston, (1966)).

[15] Dantzig, G.B., "Determining Optimal Policies for Ecosystems," IIASA, Laxenburg, Austria (1974).

[16] Dzielinski, B.P., and R. Gomory, "Optimal Programming of Lot Sizes, Inventory and Labor Allocations," Management Science, 11, No. 9, (1965).

[17] Carter, R.P., Zimmerman, R.E. and A.S. Kennedy, "Strip Mine Reclamation in Illinois," Report prepared for Illinois Institute for Environmental Quality, December (1973).

APPENDIX 1

EXAMPLE FOR TIMBER RESOURCE

We shall introduce the procedure for developing and specifying management alternatives, within a specific and simplified context. We have chosen to consider for purposes of illustration, a resource in which the primary product derived from the land is timber. This will serve to introduce the reader to the formal model outlined in Section 2. In devising this example we have been strongly influenced by the Timber RAM approach to the timber management problem [1,2].

The resource is assumed to be specified initially as a set of timber *resource classes*, each class being approximately homogeneous with respect to its silvi-cultural and economic characteristics. We shall assume for simplicity, that we have a single timber resource class, of a given size -- measured in some units of area, say thousands of acres (TA). Assume also that there is only a single age class of initial standing timber of average age A decades, and that the density of the initial standing timber is d_1 board feet per thousand acres (BF/TA). In arriving at this figure a good deal of averaging is involved, since the assumption of homogeneity is, in practice, far from true.

A plan for the management of this class must be drawn up, from the present time up to a planning horizon P decades hence -- the *planning period*: A simple management plan would establish guidelines on how much timber to remove during each decade. In order to avoid,

in this illustration, the complications arising from the need for reforestation once a stand has been clear-cut (meaning all standing timber is removed), we shall assume that no part may be clear-cut before the end of the planning period. Only during the last interval is clear-cutting allowed.

A sequence of management or control decisions carried out on any portion of the resource class, and the impact of each decision on the land, may be the following:

Time	State of the land	Control decision	Cost of decision	Benefit derived
Beginning of first decade	$\tilde{d}_1$ BF/TA of standing timber	Partial cut to level $\hat{d}_1$ BF/TA.	$\tilde{c}_1$ \$/TA	$\tilde{b}_1$ BF/TA
Beginning of second decade	Timber has grown from level $\hat{d}_1$ to level $\tilde{d}_2$ BF/TA.	Partial cut to level $\hat{d}_2$ BF/TA.	$\tilde{c}_2$ \$/TA	$\tilde{b}_2$ BF/TA
⋮	⋮	⋮		
Beginning of i^{th} decade	Timber has grown from level $\hat{d}_{i-1}$ to $\tilde{d}_i$ BF/TA.	Partial cut to level $\hat{d}_i$ BF/TA.	$\tilde{c}_i$ \$/TA	$\tilde{b}_i$ BF/TA
⋮				
Beginning of p^{th} decade	Timber at level $\tilde{d}_p$ BF/TA.	Clear cut.	$\tilde{c}_p$ \$/TA	$\tilde{d}_p$ BF/TA
End of Planning period.	No standing timber.	--	--	--

All cuts above are assumed to be instantaneous and to take place at the beginning of a decade; then the woodland is allowed to develop under the influence of the normal ecological conditions governing its behavior, until the beginning of the following decade, when another cut may be initiated.

We may depict such a sequence of control decisions as a path under a growth curve. The growth curve specifies the approximate levels of standing timber at different periods of time, in the absence of any timber harvesting.

Note that the sequence of decisions does not specify how much of the class it must manage, i.e., costs and benefits are stated on a per TA basis. For the inherent assumption of linearity to hold good, management would normally be undertaken on a fairly large scale.

Now it is clear that potentially there exists an infinite set of such decision sequences, each representing a different management alternative. We therefore discretize the state space by imposing the restriction that, at the beginning of any interval i, any part of the class may exist only at certain timber density levels, denoted by $d_{i1}, d_{i2}, \ldots, d_{ij}, \ldots, d_{in_i}$ BF/TA. These define the n_i *states* at the start of interval i, each state being characterized by the level of the *parameters* average timber density and average age of standing timber. With these parameters therefore we abstract certain essential characteristics of the timber class from the complex physical state in which any portion may exist, namely, those that are considered,

for the purpose of management, to adequately describe the productive level of the timber class. The age, which is deducible directly from i, is given by $A + i - 1$ (where recall that A denoted the age of the timber class at the beginning of the first interval). We shall not therefore mention age explicitly below. The age parameter does become significant if we allow reforestation during any interval.

Consider two states defined by the density parameter levels d_{ip} and $d_{(i+1)q}$, at the beginning and end of the i-th interval, respectively. If acreage in the state corresponding to density level d_{ip} at the beginning of the i-th decade can be cut down to a certain level $\hat{d}_{ip}$ (in our simple model all cuts are assumed to be instantaneous and the model is deterministic), and $\hat{d}_{ip}$ under normal conditions then grows back to approximately level $d_{(i+1)q}$ by the end of the i-th decade, then we say the state corresponding to d_{ip} can be *transformed* into state $d_{(i+1)q}$. The cost of such a transformation is obtained from a knowledge of the volume of timber removed and the cost of the control operations involved in the harvesting. The benefits, i.e., the amount of timber harvested may be calculated from the state parameter levels and is given by $(d_{ip} - \hat{d}_{ip})$. A rule may also exist for deducing the cost from the level of the parameters involved in the transformation and some basic information about the control operations.

If we now draw a graph of all transformations from states at the beginning and end of the i-th interval we obtain a bipartite graph. Missing arcs simply that no transformation exists between the corresponding pairs of states, or if one exists, it is not desirable. For

each interval we have such a bipartite graph, and displaying them simultaneously results in a network whose nodes are the states and whose arcs are the possible and desirable state transformations. Each arc specifies the cost and benefit of the corresponding state transformation and each path from the beginning to an end state determines a sequence of control decisions and their associated costs and benefits, i.e., a management alternative.

The reader should be aware of some of the assumptions underlining the above procedure. We have discussed resource class classification and linearity already. Another consideration is that the level of standing timber on any acre in decade i will, in reality, be dependent on the complete sequence of cuts during *all* preceding decades. In the above model this has not been taken into account. Previous successful models, e.g., Timber RAM [1] have not taken this into consideration either and therefore this aspect of our model should not represent an unacceptable departure from silvicultural reality. Whether it does, however, is a question that must be answered by a person with special skills in forestry. The situation may be rectified by introducing further parameters to describe the state of the land. However our main purpose in this section, is to illustrate the formal model, and not to explore its application to a specific resource in any great detail, so we shall defer further discussion of these points.

It has already been emphasized that the network does not specify the level of a management alternative, i.e., the number of acres that it manages. This is determined by mathematical programming as discussed in the main body of this report.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER SOL 78-31	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A LAND MANAGEMENT MODEL USING DANTZIG-WOLFE DECOMPOSITION		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER SOL 78-31
7. AUTHOR(s) L. Nazareth		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research -- SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research Program -- ONR Department of the Navy 800 N. Quincy Street, Arlington, VA 22217		12. REPORT DATE December 1978 13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This report has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Land Resource Management Modeling
Decomposition Methods
Network Flow Algorithms
Forest Management
LP Solution of Land Resource Management Models
Dynamic Programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 78-31
A LAND MANAGEMENT MODEL USING DANTZIG-WOLFE DECOMPOSITION
L. Nazareth

This paper deals with a mathematical model designed to provide guidelines for managing a land resource over an extended period of time.

We develop a framework which permits sequences of management decisions to be conveniently formulated, and their associated costs and benefits specified This takes the form of a network. Each path in the network represents a possible decision sequence. We study how to select suitable decision sequences and what proportion of the resource to manage with each selected sequence, so as to optimize some specified objective and meet the constraints imposed on management of the resource. An L.P. model is formulated. The solution strategy decomposes the L.P. matrix using Dantzig-Wolfe decomposition and solves the subproblems efficiently by dynamic programming or a network flow algorithm. Computational aspects are discussed and the concepts and procedures are illustrated in the Appendix, for forest management.

This paper is a substantially revised version of Reference [6].